EXPLORE
Jurnal Sistem Informasi dan Telematika
(Telekomunikasi, Multimedia dan Informatika)

Perancangan Aplikasi E-Menu Restaurant dengan Menggunakan Cloud Computing dan Serverless Architecture Lambda

Dian Oktaviani, Frederik Samuel Papilaya, Penidas Fiodinggo Tanaem
Program Studi Sistem Informasi
Universitas Kristen Satya Wacana
Salatiga, Jawa Tengah, Indonesia
682019603@student.uksw.edu, samuel.papilaya@uksw.edu, penidas.fiodinggo@uksw.edu

Abstract-The global competition of international restaurants has taken place in Indonesia. This competition caused the restaurants to continuously increase their service quality. Moreover, due to to the massive amount of information on the internet, E-Menu system is expected to ease customers in online purchases. The system is built through utilizing cloud computing and serverless architecture lambda. With cloud computing, accessing the cloud become easier, flexible, and quickly scalable. While serverless lambda is a method to make applications without managing the infrastructure, thus reducing expenses. The purpose of designing a restaurant E-Menu with the use of serverless lambda is to provide a modern and efficient service system for customers and the use of serverless lambda without server management so as to minimize costs because the payment of fees is only in accordance with the computing time used. This research produces E-Menu design by utilizing cloud computing technology based on serverless architecture Lambda to provide useful benefits for restaurants and customers in the fields of ordering and transactions.

Keywords: Designing, Cloud Computing, Serverless Lambda

Abstrak- Persaingan global di mana restoran dari mancanegara sudah masuk ke Indonesia tentu membuat pihak restoran akan selalu berusaha meningkatkan kualitas sistem pelayanan. Di samping itu mengingat banyak sekali informasi yang didapat melalui internet, maka bila dengan adanya sistem informasi E-menu restoran diharapkan dapat memberi kemudahan kepada pelanggan untuk melakukan pembelian secara online. Sistem ini dibangun dengan memanfaatkan teknologi Cloud Computing dan serverless architecture lambda. Dengan menggunakan cloud computing, akses cloud sangat mudah, fleksibel, dan skalibilitas yang cepat. Sedangkan menggunakan serverless lambda merupakan cara untuk membangun aplikasi tanpa perlu mengelola infrastructure sehingga akan mengurangi pengeluaran biaya. Tujuan perancangan E-Menu restaurant dengan penggunaan serverless lambda adalah memberi sistem pelayanan yang modern dan efisien bagi pelanggan serta penggunaan serverless lambda yang tanpa pengelolaan server sehingga meminimalisir biaya karena pembayaran biaya hanya sesuai dengan waktu komputasi yang digunakan. Pada penelitian ini menghasilkan peracangan E-Menu dengan memanfaatkan teknologi cloud computing berbasis architecture serverless lambda untuk memberikan manfaat yang berguna bagi pihak restoran maupun pelanggan dalam bidang pemesanan dan transaksi.

Kata Kunci: Perancangan, Cloud Computing, Serverless Lambda

1. Pendahuluan

Dengan internet banyak sekali informasi yang dengan mudahnya diakses secara cepat dan mudah, sehingga sebagian besar kegiatan manusia selalu bersinggungan dengan teknologi yang sudah ada [1]. Peran teknologi dapat mempermudah segala kegiatan yang berhubungan dengan pengelolaan dan penyajian informasi data denga cepat dan lebih akurat. Pada persaingan global saat ini memicu datangnya beragam restoran dari mancanegara yang masuk ke Indonesia. Hal ini dapat menimbulkan persaingan tidak hanya dalam hal makanan dan tempat yang megah, namun kualitas sistem pelayanan jasa harus menjadi pertimbangan pihak restoran untuk memahami kebutuhan pelanggan terhadap sistem pelayanan yang diberikan [2].

Dengan bertambahnya jumlah restoran yang tersebar di wilayah Indonesia dan semakin berkembangnya teknologi, wajib bagi restoran untuk mengembangkan dalam hal pelayanan salah satunya dengan menggunakan E-Menu. E-Menu dapat dikatakan sebagai daftar menu elektronik sebuah restoran yang dapat diakses menggunakan barang elektronik yang fungsinya sebagai metode pemesanan konvensional dan dapat lebih mempersingkat waktu serta meningkatkan kualitas pelayanan [3]. *Cloud Computing* menjadi teknologi yang

berkembang saat ini, informasi yang diolah akan lebih efisien dan semua sistem penyimpanan yang dapat diakses melalui Internet [4].

Pada penelitian sebelumnya untuk membangun perancangan E-menu restoran adalah perancangan E-Menu berbasis website dan android yang bertujuan untuk mempermudah proses transaksi yang ada dengan memanfaatkan teknologi elektronik seperti *smartphone* atau komputer dan tidak lagi dengan cara manual seperti pencatatan pesanan pada kertas menggunakan pena dan sebagai penggantinya mengarah pada media digital [1]. Hasil kesimpulan dari penelitian tersebut adalah persamaan dalam penelitian yaitu membahas tentang perancangan aplikasi E-menu restaurant, dan dari penelitian tersebut dijelaskan mengenai tujuan dari membangun sebuah aplikasi E-Menu yang telah dirancang sehingga peneliti meneliti mengenai membangun aplikasi berbasis website dan android, sedangkan penulis mengambil topik yang berbeda yaitu tentang perancangan aplikasi E-menu restoran dengan pemanfaatan teknologi *Cloud Computing* dan serverless architecture.

Karena mengingat banyaknya para pengguna yang cukup beragam menggunakan E-Menu, maka dibutuhkan sistem untuk menyimpan dan mengelola data restoran tersebut dengan penyimpanan yang cukup besar melalui internet. Platform layanan *Cloud Computing* mengarah pada perkembangan teknologi saat ini. Client dapat mengakses data yang dia butuhkan dengan menggunakan internet. Berdasarkan penelitian ini sistem E-Menu restoran dirancang dengan memanfaatkan teknologi *Cloud Computing* dan *serverless architecture*.

Selain itu dengan menggunakan *Cloud Computing* pengguna akan mendapatkan banyak keutungan diantaranya akses cloud yang mudah dan fleksibel, skalabilitas yang cepat, serta layanan (Cloud tanpa investasi awal) atau sebagai pengguna, dalam membangun *Cloud Computing* tidak perlu membayar biaya investasi yang terlalu besar, penyedia layanan cloud pengguna dapat menggunakan perangkat komputer, perangkat jaringan, dan lainnya sehingga tidak ada biaya pembelian dan juga mengurangi biaya managemen IT [5]. Sedangkan menggunakan serverless architectute atau arsitektur tanpa server merupakan langkah untuk merancang dan mengeksekusi aplikasi tanpa perlu pengelolaan infrastruktur hal tersebut tentu akan mengurangi biaya pengeluaran dan menguntungkan developer dalam penghemat waktu dalam pengembangan sebuah produk [6]. Dalam penelitian ini bertujuan untuk membangaun rancangan aplikasi E-Menu restoran berbasis *Cloud Computing* dengan pemanfaatan *serverless architecture* lambda yang bertujuan untuk memudahkan komunikasi antara pihak restoran dengan pelanggan dalam hal pemesanan menu dan pembayaran. Serta membuat rancangan aplikasi E-Menu restoran dengan biaya yang cukup hemat dengan menggunakan *Cloud Computing* dan *serverless architecture* lambda.

2. Dasar Teori

A. Cloud Computing

Cloud Computing adalah gaya baru komputasi untuk skala dinamis dan sumber daya virtual disediakan melalui internet. Manfaat dari *Cloud Computing* adalah untuk pengurangan biaya komputasi, serta mempercepat skalibitas yang saat ini mendukung bagi dunia industri [7]. Layanan dari *Cloud Computing* dibagi menjadi tiga bagian: Cloud Software as a Service (SaaS) atau provider yang keunggulannya adalah, pengguna tidak perlu memikirkan lisensi software, software yang tersedia dapat digunakan dimanapun dan kapanpun oleh pengguna. Cloud Platform as a Service (PaaS) atau provider yang keunggulannya untuk mengembangkan aplikasi oleh pengguna. Maka dari itu pengguna mampu menggunakan aplikasi yang tersedia oleh provider tanpa harus memikirkan sistem operasi, jaringan, *database engine* dan *Cloud Infrastructure as a Service* (IaaS) atau provider yang keunggulannya bagi pengguna untuk mengkonfigurasi serta sebagai penyewa infrastruktur, seperti storage dan jaringan secara virtual dan dapat mengubah *scale up* atau *scale down*. [8].

B. Serverless Architecture

Severless Architecture atau arsitektur tanpa server merupakan langkah untuk membangun dan menjalankan berbagai aplikasi dan layanan tanpa perlu mengelola infrastruktur.Server masih menjalankan suatu aplikasi, tetapi semua manajemen server dilakukan oleh penyedia layanan. Sehingga tidak lagi perlu menyediakan atau mengelola server untuk menjalankan aplikasi, database, dan sistem penyimpanan. Dengan komputasi tanpa server, tugas manajemen infrastruktur seperti penyediaan kapasitas dan patching ditangani oleh penyedia layanan, sehingga hanya dapat fokus pada penulisan kode yang melayani pelanggan. Dengan demikian hal ini dapat mengurangi biaya pengeluarab tambahan dan juga dapat menghemat waktu dan energi bagi developer [6].

C. AWS Lambda

AWS Lambda adalah suatu layanan dimana pengguna dapat menjalankan kode tanpa memerlukan atau mengelola server. Dan dengan Lambda biaya yang dibayar sesuai dengan waktu komputasi yang gunakan. Lambda dapat digunakan oleh pengguna untuk menjalankan kode hamper untuk semua jenis aplikasi atau layanan backend tanpa administrasi. Secara otomatis AWS Lambda akan menjalankan kode pengguna tanpa menyediakan atau pengelolaan server dan menskalakan aplikasi pengguna dan menjalankan kode sebagai respons pada tiap pemicu. Hanya dengan menuliskan kode dan mengunggah ke Lambda, kode pengguna akan bekerja secara paralel dan melakukan proses pada setiap pemicu secara individual, serta menskalakan secara tepat dengan ukuran beban kerja.

Dengan menggunakan *serverless architecture* lambda, pengguna dapat menulis coding pada form function sesuai dengan bahasa pemograman yang dipakai dan fungsi tersebut dapat berjalan pada cloud dan mengunggahnya pada layanan *Cloud Computing* dan dapat dipanggin sebagai respon terhadapt event yang berbeda. Seperti peristiwa upload file, update create database, dan lain – lain [6]

D. Amazon API Gateway

Amazon API Gateway penyedia adalah layanan dengan pengelolaan yang baik dan dapat mempermudah pengembangan dalam membuat, menerbitkan, memelihara, memantau, dan mengamankan API pada segala skala. API Gateway berguna sebagai "pintu depan" bagi aplikasi untuk mengakses data, logika bisnis, atau fungsi dari layanan backend pengguna. API Gateway digunakan pengguna untuk membuat API RESTful dan API WebSocket yang mendukung komunikasi dua arah pada suatu aplikasi secara real time. API Gateway mendukung beban kerja terkontainer dan tanpa server, serta aplikasi web.

Semua tugas yang ditangan API Gateway terlibat dalam penerimaan dan pemrosesan hingga ratusan ribu panggilan API secara bersamaan, termasuk pengelolaan lalu lintas, dukungan CORS, otorisasi dan kontrol akses, pembatasan, pemantauan, dan pengelolaan versi API [10].

E. Amazon DynamoDB

Amazon DynamoDB adalah database multimaster dan multiwilayah yang tahan lama dan memberikan kinerja satu digit milidetik dalam skala apapun. DynamoDB dapat dikelola sepenuhnya secara aman seperti melakukan pencadangan serta pemulihan, dan caching yang terdapat dalam suatu memori aplikasi dengan skala internet.

DynamoDB yang tidak memerlukan penyediaan server atau yang disebut dengan DynamoDB tanpa server akan berhubungan dengan pengelolaan server dan patch sehingga tidak ada perangkat lunak yang harus diinstal dan dikelola atau proses pengoperasian. Secara otomatis DynamoDB akan menskalakan tabel naik dan turun untuk menyesuaikan kapasitas dan mempertahankan kinerja. Ketersediaan toleransi dalam melakukan kesalahan serta meminimalkan kebutuhan untuk perancangan kepada aplikasi pengguna [11].

3. Metodologi Penelitian

A. Metode Pengumpulan Data

Dalam mengumpulkan informasi menggunakan metode pengumpulan data dengan melakukan beberapa tahap, sebagai berikut :

1. Studi literatur

Informasi yang didapatkan untuk mengumpulkan data adalah dengan menelaah berbagai litelatur dari jurnal ilmiah melalui situs – situs internet dan mempelajari isi dari jurnal tersebut dan berbagai situs website yang berkaitan dengan kasus penelitian.

2. Observasi

Informasi yang didapatkan dalam pengumpulan data adalah dengan pengamatan dilokasi secara langsung. Contoh: melakukan pengamatan tentang bagaimana kegiatan yang terjadi di restoran yang berkaitan dengan masalah yang diteliti.

3. Wawancara

Informasi pengumpulan data yang didapat adalah dengan melakukuan wawancara secara tatap muka dengan pihak yang berkepentingan untuk mendapatkan sumber data.

B. Analisis Kebutuhan

Tujuan dalam menganalisis kebutuhan adalah untuk meminimalisir resiko kegagalan pada suatu sistem yang sedang dikembangkan. Suatu sistem dikatakan gagal apabila tidak bisa memenuhi kebutuhan yang diinginkan pengguna, atau tidak sesuai dengan proses bisnis yang berjalan. Dalam kasus ini ada 3 pengguana yang menggunakan sistem ini, diantaranya: admin restoran, koki bagian dapur, dan customer. Semua pengguna memiliki hak akses masing – masing untuk mengelola sistem. Tabel 1 menunjukkan job desc dari setiap pengguna.

Tabel 1 Analisis Kebutuhan

Hak Akses	Deskripsi
Customer	Registrasi data customer untuk melakukan login, mencari menu yang diinginkan, melakukan pemesanan dan pembayaran.
Admin	Registrasi data admin untuk melakuakan login, melakukan pengelolaan data restoran seperti menambah, ubah, dan hapus data menu, menerima pemesanan dan pembayaran dari customer, konfirmasi pemesanan ke dapur.
Koki bagian dapur	Menerima menu pemesanan, mengolah pemesanan.

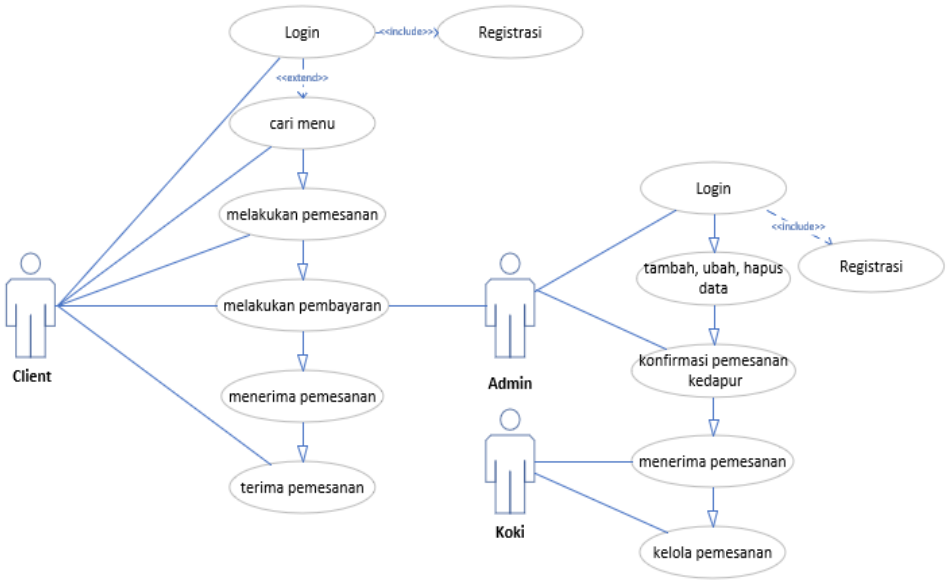
4. Hasil dan Pembahasan

A. Proses Perancangan

Perancangn telah dibuat dengan bahasa pemodelan UML, diantaranya adalah *Use Case Diagram, Activity Diagram*, dan *Class Diagram* seperti pada berikut ini:

Dalam desain sistem menggunakan metode diagram *Use Case* akan memberi sebuah narasi fungsional sebuah sistem, dan suatu program akan membutuhkan penyusunan model data yang berbentuk diagram untuk menjelaskan alur proses sistem dengan diagram *Use Case.*

1. Use Case Diagram



Gambar 1. Use Case Diagaram client, admin, dan koki

Seperti yang ditunjukkan pada Gambar 1. Use Case Diagaram client, admin, dan koki

diagram *Use Case* untuk aplikasi E-Menu restoran dimana terdapat tiga actor yaitu *client,* admin, dan koki restoran, yang masing – masing mempunyai peran untuk menjalankan kebutuhannya. Dimulai dari *client* dan Admin, agar admin mendapatkan hak akses untuk mengelola aplikai maka harus melakukan registrasi data terdahulu, untuk dapat melanjutkan ke proses login dan bisa menjalankan aplikasi sesuai kebutuhan.

Hak akses dari *client* adalah mencari menu, melakukan pemesanan dan pembayaran. Sedangkan hak akses dari admin adalah mengelola menu restoran seperti menambah, meghapus, dan mengubah data menu restoran serta mengkonfirmasikan pemesanan pelanggan ke koki bagian dapur. Koki dapur menerima daftar pemesanan dan mengelola pemesanan. **Gambar 1.** Use Case Diagaram client, admin, dan koki

2. Activity Diagram

Rangkaian proses bisnis dalam setiap *event* sistem dengan menggunakan metode diagram *activity* diagram, dengan penjelasan tentang alur proses bisnis yang terjadi pada suatu sistem, activity diagram pada aplikai E-Menu restoran ini memiliki empat *swimlane* yaitu client, admin, dapur, dan sistem yang ditunjukkan pada

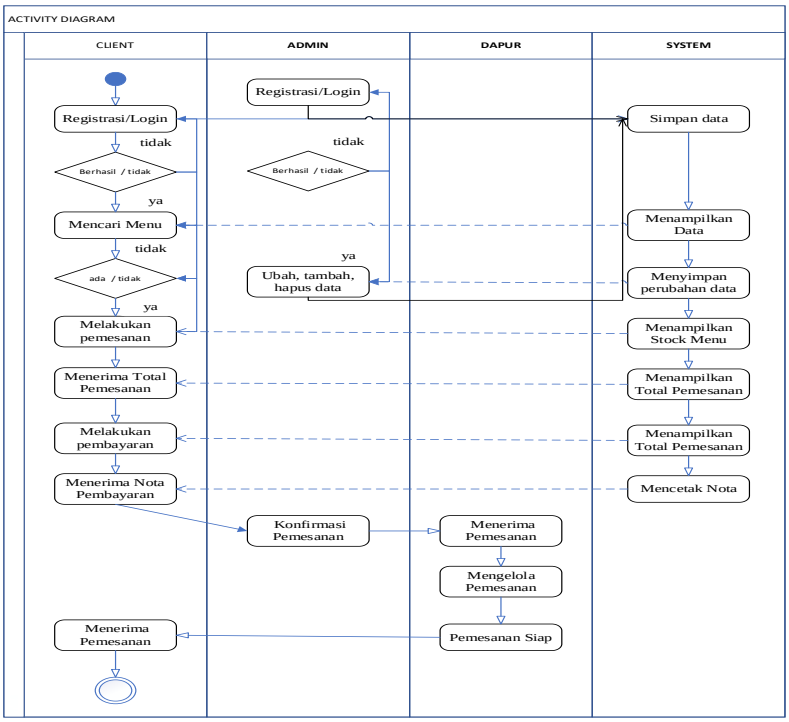
.

menunjukkan activity diagram alur proses bisnis restoran berbasis *E-menu*. Yang pertama untuk customer yang melakukan pemesanan, dimana customer melakukan registrasi dahulu untuk dapat melakukan login dan sistem akan melakukan penyimpanan data, jika login berhasil maka customer melakukan pencarian menu yang ditampilkan oleh sistem, dan dapat melakukan pemesanan dan pembayaran dan otomatis sistem menampilkan total pembayaran yang harus dibayar, jika menu tidak tersedia maka harus kembali melakukan pencarian menu.

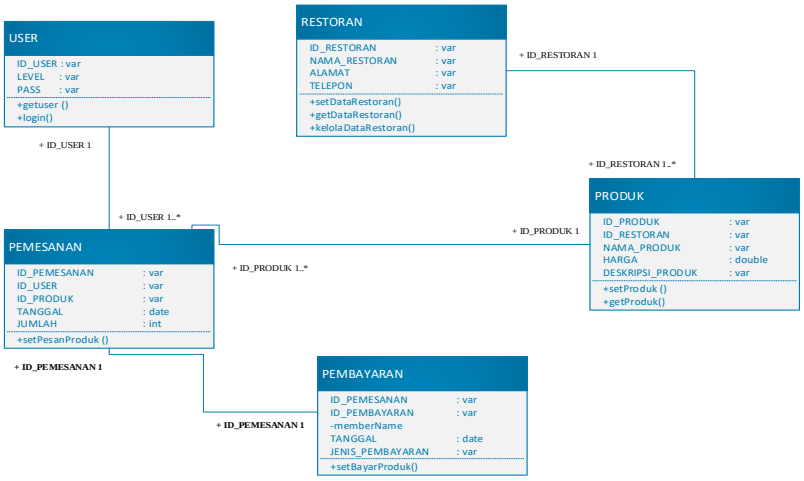
Peran admin dalam proses bisnis ini adalah mengubah, menghapus, dan menambah data, dan mengkonfirmasi pemesanan dari *client* ke dapur. Setelah menerima pemesanan dari admin bagian dapur lalu mengelola pemesanan. Dan pemesanan diterima oleh pelanggan.

3. Class Diagram

Class diagram merupakan gambaran sebuah tabel yang memiliki relasi antara satu tabel dengan yang lainnya, *Class diagram* pada aplikasi E-Menu restoran ini memiliki lima *table* yaitu user, pemesanan, restoran, dan produk, dan pembayaran yang masing – masing memiliki relasi seperti yang di tunjukkan pada



Gambar 2. Activity Diagram Alur Proses Bisnis



Gambar 3. Class Diagram E-Menu restoran

menunjukkan *class diagram* dari database *E- menu* restoran, yang masing– masing memiliki hubungan relasi dari setiap tabel. Relasi hubungan dari tabel *User* dan Tabel pemesanan adalah, satu user boleh melakukan beberapa pemesanan, sedangkan pemesanan hanya boleh mempunyai satu user. Dan dilanjutkan dengan tabel pemesanan kepembayaran, hubungan antara kedua tabel tersebut sama sama hanya memiliki satu ID_Pemesanan dan satu ID_Pembayaran.

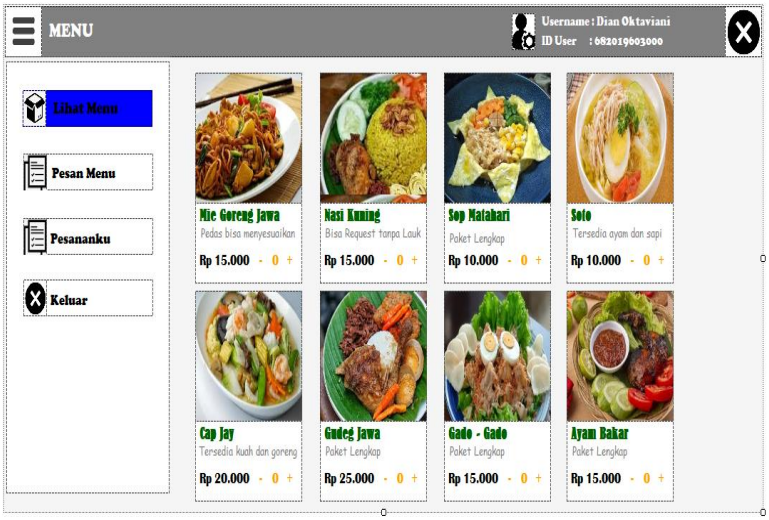
Kemudian untuk relasi hubungan antara tabel admin restoran dengan tabel produk adalah satu ID_Restoran bisa melakukan beberapa input ID_Produk ketabel produk, sedangkan ID_Produk pada tabel produk hanya satu ID_admin Restoran.

B. Implementasi User Interface

User interface untuk aplikasi E-Menu restoran di tunjukkan oleh

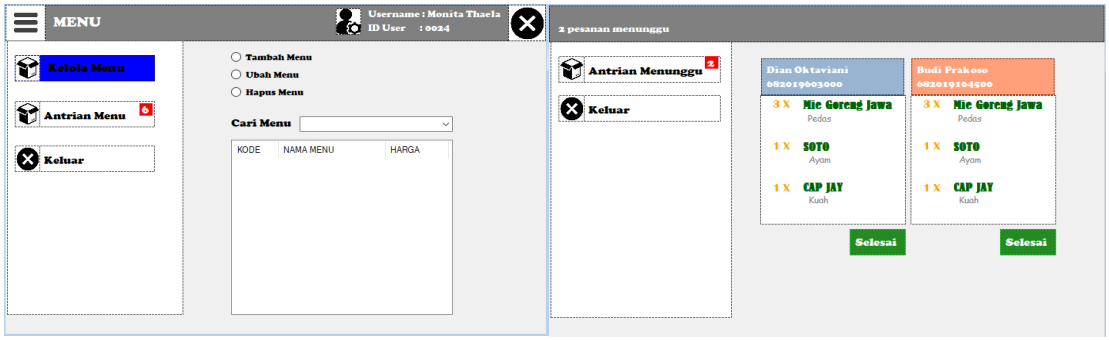
dan

dimana terdapat beberapa tampilan yang ditujukan kepada pengguna / pelanggan, admin restoran, dan koki restoran bagian dapur yang masing-masing memiliki fungsi yang dijalankan sesuai dengan kebutuhan.



Gambar 4. Tampilan untuk pengguna

menunjukkan tampilan untuk pengguna setelah melakukan regitrasi dan berhasil, pada tampilan ini tersedia identitas pengguna berupa *username*, ID pelanggan dan beberapa informasi menu seperti lihat menu yang berfungsi untuk menampilkan menu yang tersedia, pesan menu dan lihat menu mempunyai fungsi yang sama untuk melihat apa saja menu yang telah dipesan.



(A) **(B)**

Gambar 5. A) Tampilan untuk Admin restoran. B) Tampilan untuk koki restoran

Pada
menunjukkan tampilan untuk admin restoran dan koki bagian dapur. Fungsi pada

A untuk pengelolaan menu restoran berupa tambah menu, ubah menu, dan hapus menu yang dijalankan oleh admin restoran. Selain itu juga untuk memberikan informasi tentang antrian menu oleh pelanggan untuk kebutuhan dalam hal pembayaran.

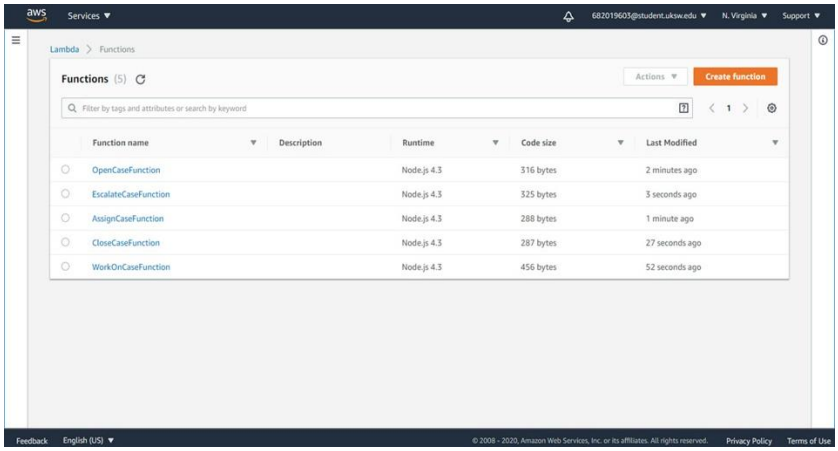
B merupakan tampilan untuk koki restoran bagian dapur yang memiliki beberapa fungsi, diantaranya untuk memberikan informasi tentang antrian menu oleh pelanggan, dan informasi menu yang dipesan oleh pelanggan sesuai dengan username dan id pelanggan[2].

C. Membuat Alur Kerja dengan Serverless Lambda

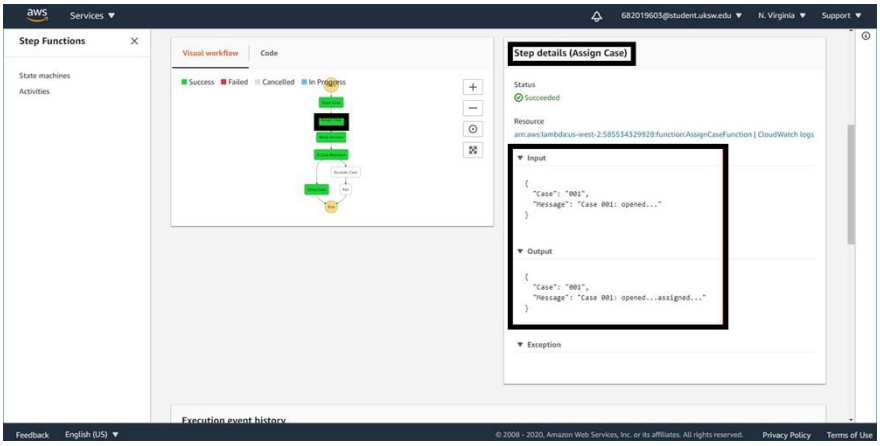
Untuk merancang alur kerja di *AWS Step Functions*, harus dikonfigurasikan dengan fungsi-fungsi *lambda* seperti yang ada pada gambar **Error! Reference source not found.**.

Gambar 6 menunjukkan 5 fungsi yang memiliki tugas masing-masing, yang pertama ada fungsi AssignCaseFunction yang bertugas menetapkan kasus dan memperbaharui pesan status, WorkOnCaseFunction yang memiliki fungsi pemberitahuan apakan kasus sudah selesai dan mengembalikan nilai bersama pesan pembaharuan, dan CloseCaseFunction untuk menutup kasus.

Kemudian saat alur kerja dieksekusi step function akan memeriksa setiap langkah, dan ID kasus yang diinputkan akan diteruskan dari setiap langkah ke langkah berikutnya. Dan akan ada pembaharuan pesan setiap fungsi lambda menyelesaikan pekerjaannya, yang ditunjukkan oleh Gambar7 [6].



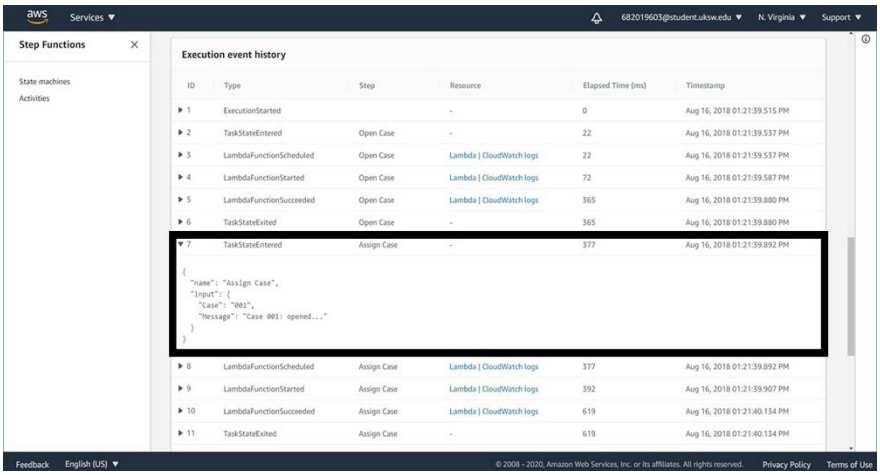
Gambar 6. Fungsi AWS Lambda



Gambar 7. Eksekusi alur kerja

menunjukkan langkah eksekusi alur kerja, termasuk input dan output setiap status. Yang awalnnya sebuah kotak dialog di isi ID untuk inputcaseID "001", yang kemudian dieksekusi. Setelah tereksekusi gulir ke bawah di bagian Riwayat eksekusi bagaimana step function memanggil fungsi lambda, seperti pada

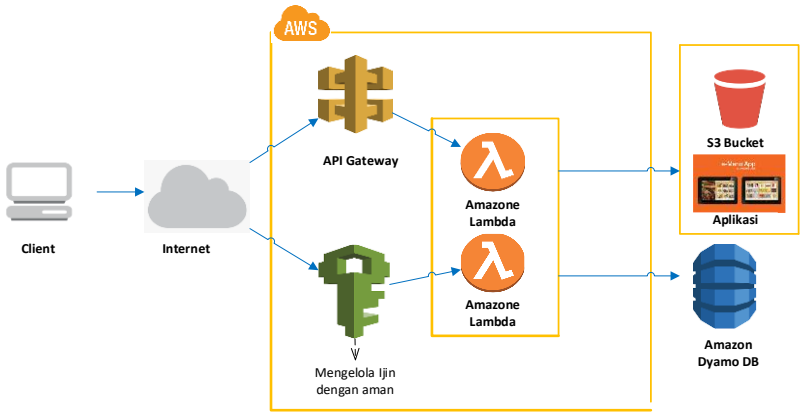
.



Gambar 8. Riwayat Eksekusi

menunjukkan bagian Riwayat kejadian eksekusi. Setiap langkah eksekusi akan terlihat bagaimana Step Functions memanggil fungsi Lambda dan meneruskan data di antara fungsi yang sudah dibuat.

D. Arsitektur Sistem



Gambar 3. Arsitektur Sistem

Gambar 8 menunjukan rancangan sistem aplikasi *E-Menu* restoran dengan *architecture serverless lambda*. Pada gambar tersebut ada *browser* sebagai *client* untuk menjalankan fungsi aplikasi tertentu. Antarmuka aplikasi yang telah dibangun akan terkonfigurasi dengan *Amazon Simple Storage Service (S3)* untuk meng-hosting sumber daya statis aplikasi web. *Web Service* pada penelitian ini menggunakan platform *Serverless AWS Lambda*. Kemudian untuk menambahkan fungsi dinamis maka digunakan Amazon API Gateway untuk memanggil API RESTful jarak jauh [9]. REST atau *'Representational State Transfer'* adalah transfer Status Representasi yang menggambarkan pola arsitektur dalam pembuatan layanan web. Sedangkan API atau application program interface adalah antarmuka dalam suatu program aplikasi. Maka demikian API RESTful yang akan mengimplementasikan pola arsitektur tersebut [12].

Kemudian layanan *AWS Identity and Access Management (IAM)* dibutuhkan untuk memberikan izin secara aman dan diperlukan untuk layanan yang saling berinteraksi [12]. Pemrosesan data yang sudah tersimpan akan tamping oleh *dynamoDB* [10]. Pada penelitian ini sistem *E-Menu* dibangun dengan *AWS Lambda.*

5. Kesimpulan dan Saran

A. Kesimpulan

Pada penelitian ini, dihasilkan beberapa perancangan dan prototipe aplikasi seperti tampilan untuk pengguna, tampilan kebutuhan untuk pengelolaan menu oleh admin, dan tampilan antarmuka oleh bagian dapur. Selain dari desain tampilan E-Menu tersebut juga dilengkapi dengan desain sistem diagram Use Case, diagram Activity, dan diagram Class yang berfungsi untuk membaca proses bisnis aplikasi. Hasil analisa yang didapat menunjukkan sistem pelayanan yang baik adalah pelayanan yang didukung oleh metode pelayanan yang meliputi proses pemesanan menu oleh peanggan, pengelolaan kategori oleh admin, pembayaran, dan semua sistem yang terintegrasi.

Arsitektur *Cloud Computing* berbasis serverless lambda mampu menjadi perancangan yang baik untuk membangun aplikasi E-Menu Restoran, sehingga tingkat skabilitasnya mampu mencangkup sistem pemasaran global, selain itu dengan menggunakan serverless lambda mengurangi biaya management IT.

B. Saran

Saran yang dapat diberikan mengenai penelitian ini yaitu berkaitan terhadap pengujian antarmuka yang perlu dilakukan, untuk menilai kelayakan aplikasi tersebut sudah terbukti efektif untuk pengguna khususnya pelanggan restoran. Dan diharapkan aplkiasi E-Menu ini dapat diterapkan pada restoran – restoran yang ada di khusunya kota Salatiga supaya meningkatkan pelayanan.

6. Daftar Pustaka

[1] Martono, ""Pembuatan Aplikasi E-Menu (Electronic Menu) Berbasis Website dan Android"," JURNAL ILMIAH MEDIA SISFO , vol. 12, p. 1036, 2018 .

[2] V. W. K. Muda, ""Rancangan Bangun Aplikasi Booking Restorant Folks Surabaya Berbasis Web"," p. 1, 2018 .

[3] Hendri, ""PROTOTIPE APLIKASI PEMESANAN MAKANAN,"," Jurnal Ilmiah Media Processor, p. 622, 2016.

[4] C. Marinescu, Cloud Computing Theory and Practise, USA: Morgan Kaufmann, 2013.

[5] LAVINDA, ""Cloud Computing dan Perannya Dorong Pertumbuhan,"," 14 JULI 2010. [Online]. Available: LAVINDA, "Cloud Computing dan Perannhttps://www.jurnal.id/id/blog/cloud-computing-dan-perannya-dorong-pertumbuhan-bisnis-anda. [Accessed 24 NOVEMBER 2020].

[6] ""Membangun Aplikasi dengan Arsitektur Tanpa Server,"," Amazon Web Service, [Online]. Available: https://aws.amazon.com/id/serverless/. . [Accessed 2020 NOVEMBER 24].

[7] A. L. &. W. A. Triyanto, ""ANALISA DAN PERANCANGAN SISTEM PEMASARAN UMKM TERINTEGRASI BERBASIS,"," SIMETRIS, vol. 5, p. 2, 2014.
[8] E. Riana, "Implementasi Cloud Computing Technology dan Dampaknya Terhadap Kelangsungan Bisnis Perusahaan Dengan Menggunakan Metode Agile dan Studi Literatur," JURIKOM (Jurnal Riset Komputer), vol. 7, 2020.
[9] "AWS Identity and Access Management," Amazone Web Service, [Online]. Available: https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html. [Accessed 23 DESEMBER 2020].
[10] "Amazon API GATEWAY," AMAZON WEB SERVICE, [Online]. Available: https://aws.amazon.com/id/api-gateway/#:~:text=Amazon%20API%20Gateway%20adalah%20layanan,fungsi%20dari%20layanan%20backend%20Anda.. [Accessed 30 NOVEMBER 2020].
[11] "Amazon DynamoDB," Amazon Web Service, [Online]. Available: https://aws.amazon.com/id/dynamodb/. [Accessed 30 NOVEMBER 2020].
[12] "Modul 1: Membuat sebuah Situs Web Statis," Amazone Web Service, [Online]. Available: https://aws.amazon.com/id/getting-started/hands-on/build-web-app-s3-lambda-api-gateway-dynamodb/module-one/. [Accessed 31 12 2021].